The Thesis Committee for Vikram Ramanathan
Certifies that this is the approved version of the following thesis:

Control of a Wheeled Mobile Robot with Centered Orientable Wheels and an Offset Alpha Sensor for Radiation Surveying Applications

APPROVED BY

SUPERVISING COMMITTEE:

Control of a Wheeled Mobile Robot with Centered Orientable Wheels and an Offset Alpha Sensor for Radiation Surveying Applications

by

Vikram Ramanathan

UNDERGRADUATE HONORS THESIS

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2020

Acknowledgments

First of all, I would like to thank my faculty advisor, Dr. Mitchell Pryor for all his invaluable support, guidance and advice during this formative research experience. His willingness to share his vast repository of experience in the field of robotics while actively encouraging me to explore and develop my own ideas has helped me grow both as a researcher and an engineer. I am also very fortunate to have benefited from the mentorship of my second reader, Dr. Luis Sentis, who has consistently motivated and challenged me to develop sound foundational knowledge in several mathematical and control-theoretic principles. Dr. Sentis has not only taken active interest in my research for this thesis project, but also provided invaluable guidance and resources to foster my development in the field of humanoid robotics.

I would also like to extend my gratitude to Andy Zelenak and Adam Pettinger for their insights on hardware, software and controls-related problems I have faced along the way. Their advice has advanced my knowledge of the nuances of robot development, maintenance and troubleshooting.

None of this could have been done without all the love and support of my family. I would like to thank my parents for everything they have done support my growth and all their sagely advice. Also, to my younger brother, Vivek - thank you so much for putting up with me and keeping me sane.

Finally, I would like thank the Texas Exes and the entire scholarship staff for

making my time as a Longhorn and Forty Acres Scholar extremely memorable and rewarding.

Abstract

Control of a Wheeled Mobile Robot with Centered Orientable Wheels and an Offset Alpha Sensor for Radiation Surveying Applications

Vikram Ramanathan, B.S.M.E.
The University of Texas at Austin, 2020

Supervisors: Mitch Pryor
Co-Supervisor: Luis Sentis

This thesis considers the modeling, control and path planning of wheeled mobile robots with four Centered Orientable Conventional (COC) wheels, intended to assist with alpha radiation surveys. When compared to non-conventional wheels, COC wheels perform better over rough terrain, are not subject to vertical chatter and offer better braking capability. However, COC wheels are pseudo-omnidirectional and subject to nonholonomic constraints. Several established modeling and control techniques define and control the Instantaneous Center of Rotation (ICR); however, this method involves singular configurations that are not trivial to eliminate. This work proposes a method that uses a novel ICR-based kinematic model to avoid these singularities, and an ICR-based nonlinear controller for one 'master' wheel. The other 'slave' wheels simply track the resulting kinematic relationships between the 'master' wheel and the ICR. Thus, the nonlinear control problem is reduced from 12^{th} to

3^{rd}-order, becoming much more tractable. Simulations with a feedback linearization controller verify the approach.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Omnidirectionality has become a vital feature for Wheeled Mobile Robots (WMRs) employed in cluttered, hazardous and highly dynamic environments. The increased maneuverability of omnidirectional WMRs (OWMRs) allows for a high degree of flexibility in the planning of motion-efficient paths through free space. The majority of well-established design, modeling, and control techniques for OWMRs [15] [14] involve the use of (i) non-conventional wheels, such as Mecanum wheels and omnidirectional wheels, and/or (ii) orientable conventional wheels.

Non-conventional wheels are subject to highly complex mechanics [11] and are consequently difficult to model accurately. In addition, these wheels tend to vibrate vertically, have limited load capacity [10], generate undesirable swerving motions during braking [7], and cannot traverse rough terrain.

Orientable, or steered, conventional wheels are omnidirectional and subject to holonomic constraints if and only if there is a non-zero steering axis offset [2]. Hence, since Centered Orientable Conventional (COC) wheels - by definition - have no steering axis offset, they are pseudo-omnidirectional and subject to nonholonomic constraints. This is due to the fact that a reorienting steer command must be exe-

cuted for each wheel before a specific robot velocity and orientation can be attained. Thus, the precise coordination of wheel drive and steer commands for such WMRs is necessary to avoid slip and damage to the wheel modules. Campion [1] established that for WMRs with more than 2 COC wheels, the wheels must be coordinated such that the degree of steerability, δ_s, is 2. Here, δ_s is the number of COC wheels that can be oriented independently in order to steer the robot. This implies coupling of particular wheel motions and redundancies in the control of WMR wheel modules. Muir and Neuman [15] also note that COC wheels have singular wheel Jacobian matrices. Therefore, the soluble motion criterion is not satisfied, and Muir and Neuman's proposed actuated inverse solution cannot be used for the control of WMRs with COC wheels.

Thus, the modeling and control of WMRs with COC wheels is challenging. One approach is to coordinate each wheel's drive speed and orientation with respect to the Instantaneous Center of Rotation (ICR). The Instantaneous Center of Rotation (ICR) is defined as the point in the WMR's plane of motion about which the robot instantaneously rotates about. Thuilot [16] proved that any admissable wheel configuration can be uniquely identified by the coordinates of its associated ICR, assuming it is not located at a wheel center. More generally, ICR modeling is effective provided singular configurations are considered and avoided. There are two singularities the ICR representation is subject to. The first, stated above, is the singularity caused when the ICR is located at a wheel center. The second singularity occurs during pure translation. In this case, the ICR is at infinity. Deitrich [5] used Khatib's [8] methodology of classifying singular configurations as obstacles with repulsive potential fields.

Connette [4] represented the ICR in spherical coordinates in order to eliminate the second singularity and allow for simpler control. In a later paper [3], Connette detailed the use of repulsive potential fields around the wheel axes to avoid the first singularity and associated unbounded steering rates when the ICR passes near, but not through, the wheel steering axis.

This work presents a novel ICR-based master-slave kinematic modeling and control method for WMRs with COC wheels (see Fig. 2.2) that eliminates the singularities detailed above. Firstly, a brief overview of our WMR is provided. Secondly, we discuss the derivation of the kinematic model. Thirdly, we present a nonlinear control solution for the developed model. Lastly, we detail the simulations that showcase the validity this approach.

Chapter 2

Experimental Setup

2.1 Wheeled Mobile Robot Platform

The considered WMR (see Fig. 2.1) consists of a square aluminum frame with 4 wheel modules at each of its corners. The wheel modules used are the Swerve and Steer Modules from AndyMark Inc. Each module includes one drive CIM motor and one steer PG71 Planetary Gearbox with RS775 motor. A hall effect encoder is included on the PG71 gearmotor, and an absolute encoder is used to record the steer orientation. The wheel module's design allows the steering axis to be directly above the wheel-floor contact, i.e. there is no steering-axis offset (see Fig. 2.2).

The system design was motivated to assist with alpha radiation contamination surveys [13]. To complete radiation surveys, the sensor must be held approximately 1/4 inch above the surveyed floor and move at approximately 2 inches/second. Furthermore, as a precaution to prevent the wheels from spreading contamination, the platform must be able to survey an environment with trajectories that avoid placing the wheels in locations that have not been surveyed. Thus, a holonomic (or near holonomic) solution allows for flexibility in planning these trajectories. While our WMR lacks true holonomic control, it can be assumed to be omnidirectional at sufficiently fast steer speeds and with well-tuned low-level motor controllers.

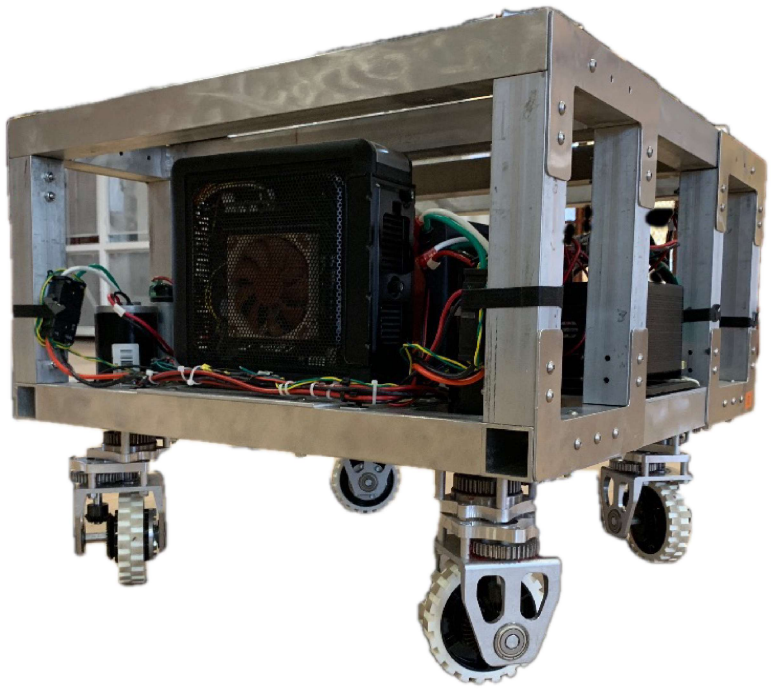


Figure 2.1: Nuclear Robotics Group's pseudo-omnidirectional WMR

Figure 2.3b depicts the platform schematically. W, and L are the robot body width and length respectively. In the inertial frame $\{0\}$ (see Fig. 2.3a), the robot's position and orientation are defined by x_r, y_r, and θ, the angle with respect to x. (x_r, y_r) can be any point on the robot's chassis; we chose the geometric center. The robot frame, $\{\mathrm{R}\}$ is defined to be fixed to (x_r, y_r) and rotates with the WMR. ${}^0\boldsymbol{m_i}$, with $i \in \{1, 2, 3, 4\}$, is an (x, y) vector in frame $\{0\}$ representing the locations of the wheel modules.

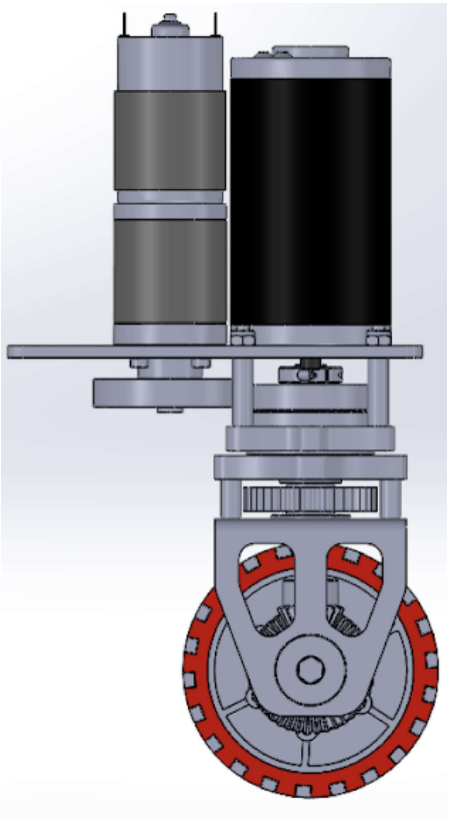
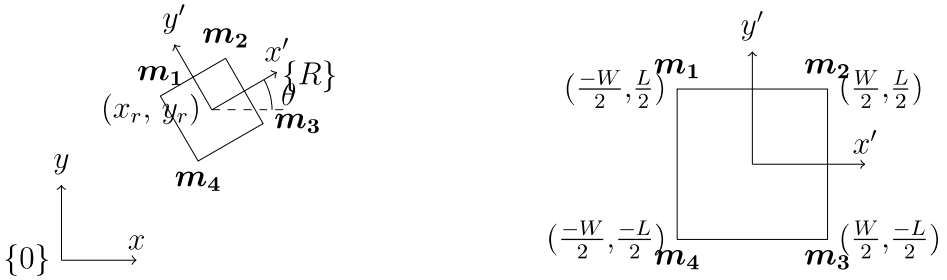


Figure 2.2: Swerve and steer module. One motor controls drive speed, another controls steer angle of the wheel.



(a) Inertial Frame, {0}

(b) Robot Frame, {R}

Figure 2.3: Coordinate frame conventions

Chapter 3

Kinematic Modeling and Control

In this chapter, we detail the proposed ICR based kinematic modeling and control structure for omnidirectional wheeled mobile robot platforms with centered orientable conventional wheels. Before presenting the feedback linearization controller and the associated control algorithm structure, we provide the derivation of the ICR kinematics and the controllability of the system.

3.1 ICR-Based Master-Slave Kinematic Modeling

The location of the ICR in the inertial frame can be expressed as [9],

$$\boldsymbol{x} = \begin{bmatrix} x_{ICR} \\ y_{ICR} \end{bmatrix} = \begin{bmatrix} x_r - \dot{y}_r/\omega \\ y_r + \dot{x}_r/\omega \end{bmatrix} \tag{3.1}$$

where $\omega = \dot{\theta}$. Let ${}^0\boldsymbol{e}$ be any vector (e_x, e_y) defined in frame $\{0\}$. It can be shown that [9],

$${}^0\dot{\boldsymbol{e}} = \omega \begin{bmatrix} -y_\rho \\ x_\rho \end{bmatrix} \tag{3.2}$$

Here, $y_\rho = e_y - y_{ICR}$ and $x_\rho = e_x - x_{ICR}$. Thus, we use Eq. 3.2 to find the velocities at the x-y positions of the wheel modules in frame $\{0\}$ (${}^0\boldsymbol{m_1}$,${}^0\boldsymbol{m_2}$,${}^0\boldsymbol{m_3}$,${}^0\boldsymbol{m_4}$):

$$ {}^{0}\dot{\boldsymbol{m}}_i = \omega \begin{bmatrix} -m_{i_y} + y_{ICR} \\ m_{i_x} - x_{ICR} \end{bmatrix} \tag{3.3} $$

where $i \in \{1, 2, 3, 4\}$. For this particular platform design, the four wheels are spaced evenly about the center of the platform. In particular, ${}^{R}\boldsymbol{m_1} = \begin{bmatrix} -W/2 \\ L/2 \end{bmatrix}$, ${}^{R}\boldsymbol{m_2} = \begin{bmatrix} W/2 \\ L/2 \end{bmatrix}$, ${}^{R}\boldsymbol{m_3} = \begin{bmatrix} W/2 \\ -L/2 \end{bmatrix}$, and ${}^{R}\boldsymbol{m_4} = \begin{bmatrix} -W/2 \\ -L/2 \end{bmatrix}$. We define the transformation from the robot to inertial frame to be,

$$ {}^{0}\boldsymbol{m_i} = \begin{bmatrix} m_{i_x} \\ m_{i_y} \end{bmatrix} = \begin{bmatrix} x_r \\ y_r \end{bmatrix} + \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} {}^{R}\boldsymbol{m_i} \tag{3.4} $$

Substituting Eq. 3.4 and Eq. 3.1 in Eq. 3.3:

$$ {}^{0}\dot{\boldsymbol{m}}_1 = \begin{bmatrix} \frac{W}{2}\omega sin(\theta) - \frac{L}{2}\omega cos(\theta) + \dot{x}_r \\ \frac{-W}{2}\omega cos(\theta) - \frac{L}{2}\omega sin(\theta) + \dot{y}_r \end{bmatrix} \tag{3.5} $$

$$ {}^{0}\dot{\boldsymbol{m}}_2 = \begin{bmatrix} \frac{-W}{2}\omega sin(\theta) - \frac{L}{2}\omega cos(\theta) + \dot{x}_r \\ \frac{W}{2}\omega cos(\theta) - \frac{L}{2}\omega sin(\theta) + \dot{y}_r \end{bmatrix} \tag{3.6} $$

$$ {}^{0}\dot{\boldsymbol{m}}_3 = \begin{bmatrix} \frac{-W}{2}\omega sin(\theta) + \frac{L}{2}\omega cos(\theta) + \dot{x}_r \\ \frac{W}{2}\omega cos(\theta) + \frac{L}{2}\omega sin(\theta) + \dot{y}_r \end{bmatrix} \tag{3.7} $$

$$ {}^{0}\dot{\boldsymbol{m}}_4 = \begin{bmatrix} \frac{W}{2}\omega sin(\theta) + \frac{L}{2}\omega cos(\theta) + \dot{x}_r \\ \frac{-W}{2}\omega cos(\theta) + \frac{L}{2}\omega sin(\theta) + \dot{y}_r \end{bmatrix} \tag{3.8} $$

Rearranging Eq. 3.5, we get the following two differential equations that express the WMR's x and y velocities as functions of θ, ω, and $\dot{m}_{1_x}$/ $\dot{m}_{1_y}$,

$$ \dot{x}_r = f(\theta, \omega, \dot{m}_{1_x}) = \frac{-W}{2}\omega sin(\theta) + \frac{L}{2}\omega cos(\theta) + \dot{m}_{1_x} \tag{3.9} $$

$$\dot{y}_r = f(\theta, \omega, \dot{m}_{1_y}) = \frac{W}{2}\omega cos(\theta) + \frac{L}{2}\omega sin(\theta) + \dot{m}_{1_y} \tag{3.10}$$

Noting that $\dot{\theta} = \omega$, we have effectively obtained three state equations with states (x_r, y_r, θ) and inputs ($\dot{m}_{1_x}$, $\dot{m}_{1_y}$, ω). Thus, a control schema can be formulated (detailed in following sections) such that in each control cycle, a nonlinear controller determines the appropriate inputs i.e. the x and y velocities for wheel module 1 in $\{0\}$ frame, ($\dot{m}_{1_x}$, $\dot{m}_{1_y}$). Next, the states and outputs from the controller can be used to determine ${}^0\dot{\boldsymbol{m}}_2$, ${}^0\dot{\boldsymbol{m}}_3$, and ${}^0\dot{\boldsymbol{m}}_4$ (using Eq. 3.6,3.7,3.8). Thus, we can use feedback control on the kinematic model for any single wheel module - in our case, we chose the module located in frame $\{0\}$ by ${}^0\boldsymbol{m_1}$ - to solve for all the required wheel module velocities in $\{0\}$ frame. We call the wheel chosen for feedback control the 'master' wheel and the remaining wheels the 'slave' wheels. By nature of the derivation of the wheel module velocities in the $\{0\}$ frame (Eq. 3.5-3.8), the slave wheels coordinate their drive speeds and steer angles to ensure their drive axes always coincide with the ICR. An added benefit of this strategy is that the nonlinear control problem becomes more tractable since it reduces the order of the controlled system from 12 to 3. While the derivation here was specific to this platform geometry, the method is extensible to many other geometries.

Once the wheel module velocities are determined in $\{0\}$ frame, we can determine each wheel module's required drive speed and steer angle, assuming no slip:

$$\psi_i = \frac{\sqrt{\dot{m}_{i_x}^2 + \dot{m}_{i_y}^2}}{r_{wheel}} \tag{3.11}$$

$$\phi_i = atan_2(\frac{\dot{m}_{i_y}}{\dot{m}_{i_x}}) - \theta \tag{3.12}$$

where ψ_i is the i^{th} wheel module drive speed, r_{wheel} is the wheel radius (same for all four wheels), and ϕ_i is the i^{th} wheel module steer angle. Thus, our chosen parameterization disallows $\omega = 0$ to be a singularity. When $\omega = 0$, the $\{0\}$ frame velocity of the master wheel and subsequently, the slave wheels, will simply be equivalent to $(\dot{x}_r, \dot{y}_r)$, resulting in pure translation.

3.2 Control

3.2.1 Controllability

To analyze controllability, from Eq. 3.9, 3.10, and $\dot{\theta} = \omega$, the system is rewritten in state-space form:

$$\boldsymbol{x} \equiv \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \equiv \begin{bmatrix} x_r \\ y_r \\ \theta \end{bmatrix}, \quad \boldsymbol{u} \equiv \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \equiv \begin{bmatrix} \dot{m}_{1_x} \\ \dot{m}_{1_y} \\ \omega \end{bmatrix}, \quad \boldsymbol{y} \equiv \boldsymbol{x} \tag{3.13}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u_2 + \begin{bmatrix} -\frac{W}{2}\sin x_3 + \frac{L}{2}\cos x_3 \\ \frac{W}{2}\cos x_3 + \frac{L}{2}\sin x_3 \\ 1 \end{bmatrix} u_3$$
$$\equiv \boldsymbol{g_1} u_1 + \boldsymbol{g_2} u_2 + \boldsymbol{g_3} u_3 \tag{3.14}$$

This matches the format of a nonlinear, input-affine system with $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \sum_{i=1}^{m} \boldsymbol{g_i}(\boldsymbol{x}) u_i \tag{3.15}$$

The accessibility distribution for this system is given by [6],

$$\boldsymbol{C} = [\boldsymbol{g_1} \quad \boldsymbol{g_2} \quad \boldsymbol{g_3} \quad [\boldsymbol{g_1}, \boldsymbol{g_2}] \quad [\boldsymbol{g_1}, \boldsymbol{g_3}] \quad [\boldsymbol{g_2}, \boldsymbol{g_3}] \quad [\boldsymbol{g_1}, [\boldsymbol{g_1}, \boldsymbol{g_2}]] \quad [\boldsymbol{g_1}, [\boldsymbol{g_1}, \boldsymbol{g_3}]] \quad [\boldsymbol{g_2}, [\boldsymbol{g_2}, \boldsymbol{g_3}]]] \tag{3.16}$$

where $[\boldsymbol{g_i}, \boldsymbol{g_j}]$ is a Lie bracket operator:

$$[\boldsymbol{g_i}, \boldsymbol{g_j}] = \frac{\delta \boldsymbol{g_j}}{\delta \boldsymbol{x}} \boldsymbol{g_i} - \frac{\delta \boldsymbol{g_i}}{\delta \boldsymbol{x}} \boldsymbol{g_j} \tag{3.17}$$

$[\boldsymbol{g_i}, \boldsymbol{g_j}] = \boldsymbol{0}$ denotes that the relative effect of $\boldsymbol{g_i} u_i$ versus $\boldsymbol{g_j} u_j$ will not vary as $\boldsymbol{x}$ changes; thus, a change in $\boldsymbol{x}$ does not yield an extra avenue to control the system. Higher-order Lie brackets are calculated recursively, i.e. $[\boldsymbol{g_i}, [\boldsymbol{g_i}, \boldsymbol{g_j}]]$ can be calculated from the result of $[\boldsymbol{g_i}, \boldsymbol{g_j}]$.

The calculation of $[\boldsymbol{g_1}, \boldsymbol{g_3}]$ below serves as an example:

$$[\boldsymbol{g_1}, \boldsymbol{g_3}] = \frac{\delta \boldsymbol{g_3}}{\delta \boldsymbol{x}} \boldsymbol{g_1} - \frac{\delta \boldsymbol{g_1}}{\delta \boldsymbol{x}} \boldsymbol{g_3} = \begin{bmatrix} 0 & 0 & -\frac{W}{2}\cos x_3 - \frac{L}{2}\sin x_3 \\ 0 & 0 & -\frac{W}{2}\sin x_3 + \frac{L}{2}\cos x_3 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - [\boldsymbol{0}] * \boldsymbol{g_3} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.18}$$

In this case, it is clear from the first three columns of $\boldsymbol{C}$ that the distribution is full rank. Thus, it is not necessary to evaluate the Lie brackets. For the sake of completeness, the end result for the accessibility distribution of this system was calculated:

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & -\frac{W}{2}\sin x_3 + \frac{L}{2}\cos x_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{W}{2}\cos x_3 + \frac{L}{2}\sin x_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.19}$$

Per Hedrick, since $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ and $rank(\boldsymbol{C}) = 3$ (from Eq. 3.19 above), this driftless system is controllable. Thus, the master wheel can theoretically be controlled to yield any desired position and orientation of the mobile platform.

3.2.2 Control Structure

A schematic of the proposed control structure is provided in Fig. 3.1. In this diagram, the NC block is the Nonlinear Controller detailed in Section 3.2.3 below, the LLC blocks are Low-Level wheel Controllers and the M_i blocks, with $i \in \{1, 2, 3, 4\}$, are the kinematic models of each wheel module. Here, the low-level controller block is an abstraction for feedback control using encoder data.

In each control cycle, we solve for the slave wheel velocities in the inertial frame, $\dot{\boldsymbol{m}}_2, \dot{\boldsymbol{m}}_3$, and $\dot{\boldsymbol{m}}_4$ using Eqs. 3.6, 3.7 and 3.8 respectively. We then use Eq. 3.11 and 3.12 to find the required steer angle and drive speed of each slave wheel. For the sake of the argument for omnidirectionality of such WMR platforms, the tuned LLCs are run at a higher frequency than the NC. This ensures that the drive speeds and steer angles prescribed by the open-loop nonlinear controller are rapidly and accurately attained.

3.2.3 Feedback Linearization Controller

In this section, we present feedback linearization as a nonlinear control technique for the input-affine and driftless system given by Eq. 3.14:

$$\dot{\boldsymbol{x}} = \boldsymbol{G}\boldsymbol{u} \tag{3.20}$$

where $\boldsymbol{G} = \begin{bmatrix} \boldsymbol{g_1} & \boldsymbol{g_2} & \boldsymbol{g_3} \end{bmatrix}$. Thus, we chose the control input to be of the form,

$$\boldsymbol{u} = \boldsymbol{G^{-1}}\widetilde{\boldsymbol{u}} \tag{3.21}$$

where $\widetilde{\boldsymbol{u}}$ is an equivalent input (i.e. $\boldsymbol{u}$ is easily determined from $\widetilde{\boldsymbol{u}}$ and vice versa). Since $det(\boldsymbol{G}) \neq 0 \ \forall \ x_3 \in \mathbb{R}$, $\boldsymbol{G}$ is globally invertible and the control law is globally defined. As a result, the nonlinearity is canceled and we obtain the obviously linear and controllable system,

$$\dot{\boldsymbol{x}} = \widetilde{\boldsymbol{u}} \tag{3.22}$$

A well-known solution for the new dynamics is the control law:

$$\widetilde{\boldsymbol{u}} = k_p(\boldsymbol{x_d} - \boldsymbol{x}) \tag{3.23}$$

where k_p is the tunable proportional gain and x_d is the vector of desired state values.

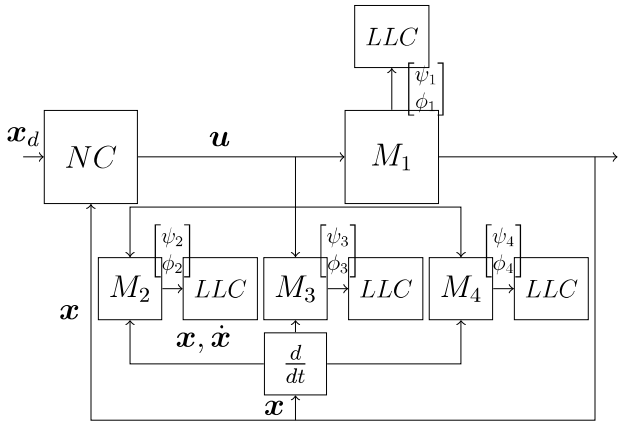


Figure 3.1: Control Structure

In the case that this controller is used in tandem with a trajectory planning algorithm that generates not only the path but also the desired velocities along the path, this controller can significantly benefit from the inclusion of a feedforward velocity term, $\boldsymbol{v_d}$:

$$\dot{\boldsymbol{x}} = \widetilde{\boldsymbol{u}} + \boldsymbol{v_d} \tag{3.24}$$

This will improve its trajectory tracking performance by suppressing unmeasured disturbances and minimizing tracking error during periods of constant velocity.

In the context of this work, the nonlinear controller can be thought of as a black box that is simply used to demonstrate the 'master-slave' kinematic modeling and control technique. Model predictive control is another viable control technique for this system.

Chapter 4

Results

4.1 Control Simulations

The proposed kinematic model and nonlinear control structure were implemented in MATLAB. Both the width and length of the WMR in our simulation are assumed to be 1 m; the wheel radius is assumed to be 0.1 m; and, the maximum wheel drive speed is assumed to be 10 rad/s.

Figure 4.1 depicts the robot motion in the $\{0\}$ frame with the arbitrarily chosen state setpoints, $x_r = y_r = 1$ m and $\theta = 1$ rad. It is evident that the robot is able to reach its desired final (green) configuration, starting from its initial (black) configuration. Figure 4.5 shows the control inputs decrease asymptotically to zero, verifying the stability of the control solution. The state response with the chosen set of target values can be seen in Fig. 4.4. Since the control law, $\widetilde{\boldsymbol{u}}$, for the linearized system dynamics in Eq. 3.24 is simply proportional control with a scalar proportional gain (given by Eq. 3.23), the three system states, x_1, x_2, x_3, share the same time response. It is also clear from Fig. 4.4 that there is neither steady-state error nor overshoot, which are highly desirable response characteristics for accurate motion planning. This is simply the result of tuning the proportional gain, k_p. It must be noted that the constant k_p may not be chosen arbitrarily. Granted that the actuators

on the robot have saturation limits (maximum drive speeds), k_p is upper bounded. Following from Eq. 3.11,

$$u_1^2 + u_2^2 \leq (r\psi_{max})^2 \tag{4.1}$$

where r is the wheel radius, ψ_{max} is the maximum wheel drive speed, and u_1, u_2 are the first and second control inputs, $\dot{m}_{1_x}$ and $\dot{m}_{1_y}$ respectively. From Eqs. 3.21 and 3.23, we solve for u:

$$\boldsymbol{u} = \boldsymbol{G}^{-1}\widetilde{\boldsymbol{u}} = \boldsymbol{G}^{-1}k_p(\boldsymbol{x_d} - \boldsymbol{x}) \tag{4.2}$$

Thus, we get the following expressions for u_1 and u_2:

$$u_1 = k_p\big[(x_{d1} - x_1) + (0.5sinx_3 + 0.5cosx_3)(x_{d3} - x_3)\big] \tag{4.3}$$

$$u_2 = k_p\big[(x_{d2} - x_2) + (-0.5sinx_3 - 0.5cosx_3)(x_{d3} - x_3)\big] \tag{4.4}$$

where x_{d1}, x_{d2}, and x_{d3} are the desired setpoints for states x_1, x_2, and x_3 respectively. Maximizing the function $h(u_1, u_2) = u_1^2 + u_2^2$:

$$h_{max} = k_p^2\big[\Delta x_1^2 + \Delta x_2^2 + \Delta x_3^2 + \sqrt{2}\Delta x_1\Delta x_3 - \sqrt{2}\Delta x_2\Delta x_3\big] \tag{4.5}$$

where $\Delta x_i = |x_{di} - x_{0i}|$, and x_{0i} is the initial condition for the ith state. Therefore, the first inequality k_p must satisfy is given by:

$$k_p \leq \sqrt{\frac{(r\psi_{max})^2}{\Delta x_1^2 + \Delta x_2^2 + \Delta x_3^2 + \sqrt{2}\Delta x_1 \Delta x_3 - \sqrt{2}\Delta x_2 \Delta x_3}} \tag{4.6}$$

Here, r and ψ_{max} are assumed to be strictly positive. Thus, $\Delta x_1^2 + \Delta x_2^2 + \Delta x_3^2 + \sqrt{2}\Delta x_1 \Delta x_3 > \sqrt{2}\Delta x_2 \Delta x_3$ must be satisfied.

The second inequality is similarly derived from Eqs. 3.6-3.8:

$$k_p \leq \sqrt{\frac{(r\psi_{max})^2}{(0.5\Delta x_3 + \Delta x_1)^2 + (0.5\Delta x_3 + \Delta x_2)^2}} \tag{4.7}$$

Once Eqs. 4.6 and 4.7 are evaluated, the lower of the two bounds is used to determine an appropriate k_p. Note that this method was developed for WMRs with square dimensions ($W = L$). Finding k_p for non-square geometries is more involved, but follows from the method detailed above.

For the WMR parameters and setpoints specified earlier in this section, $k_p \leq 0.4714$. Figure 4.2 verifies that the drive speeds of all four wheels are within the desired saturation limits. Since the plots of drive speeds (Fig. 4.2) and steer angles (Fig. 4.3) are smooth, continuous and monotonically decreasing, the WMR motion is also smooth without sudden jumps or halts. The time constant of the response is derived from Eq. 3.24 to be, $\tau = 1/k_p = 2.12s$. While faster responses are possible with larger saturation limits, they are also more susceptible to errors from the unaccounted system dynamics.

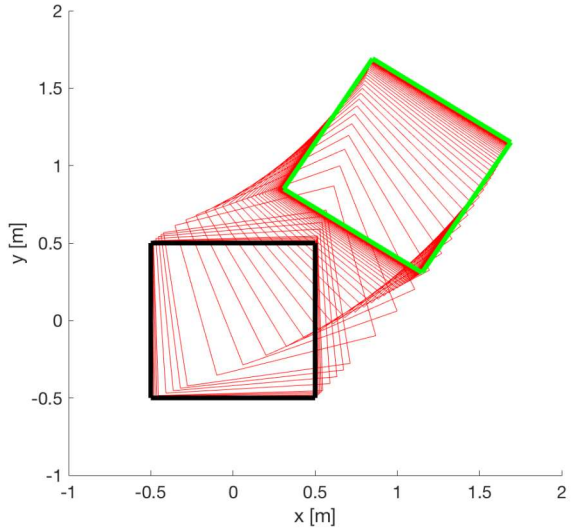


Figure 4.1: WMR motion in {0} frame

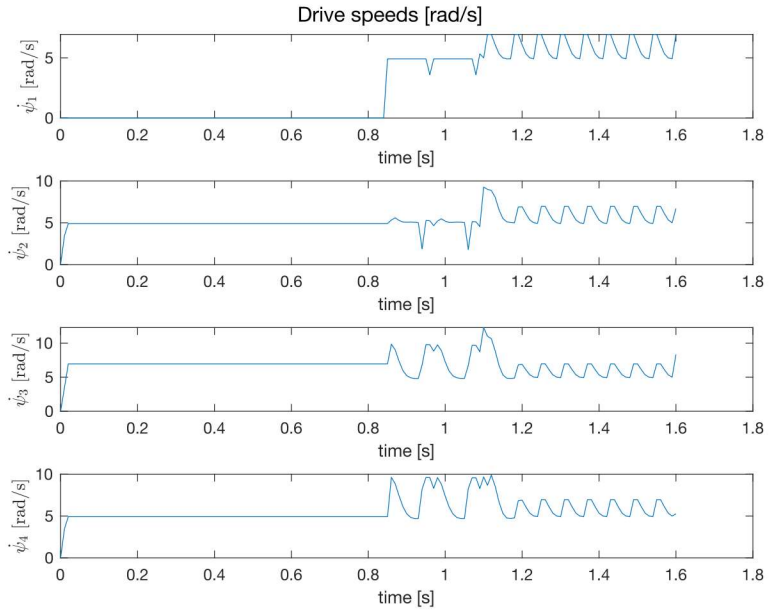


Figure 4.2: Drive speeds for each wheel module

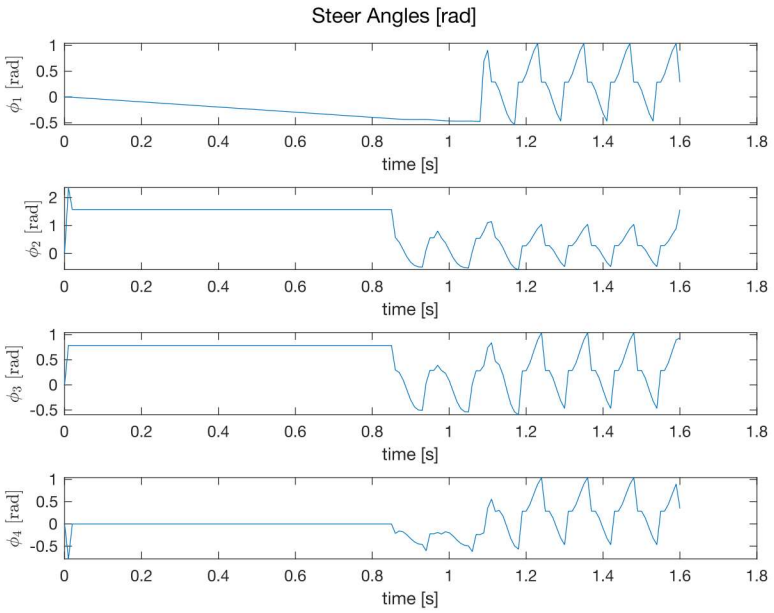


Figure 4.3: Steer angles for each wheel module

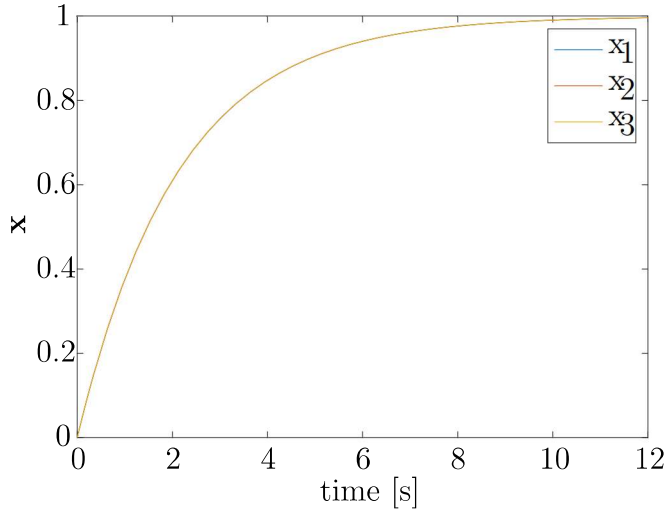


Figure 4.4: State, $\boldsymbol{x}$, response in 12 s time interval. x_1, x_2, and x_3 share the same time response

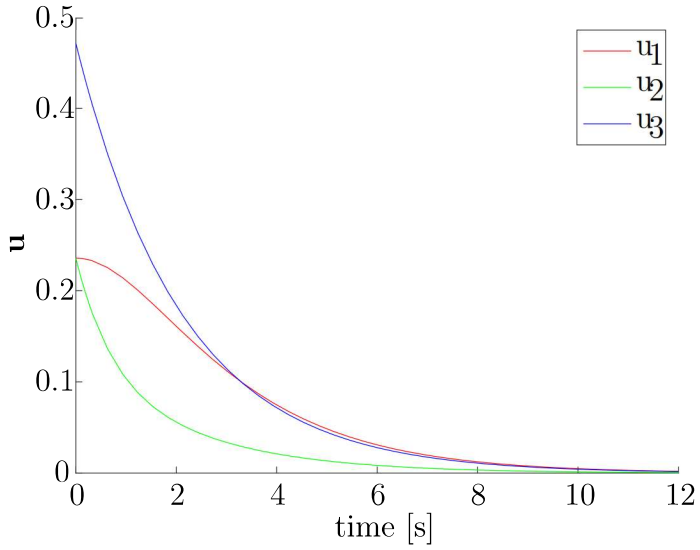


Figure 4.5: Control input from feedback linearization

4.2 Robot Simulations

Granted that access to the university campus and the robot were restricted due to the Covid-19 pandemic, this report showcases the control algorithm primarily in simulation instead of on the robot hardware. This section will present the high-fidelity simulation environment setup, the accompanying control software and finally, some example results.

4.2.1 Software Design

The robot simulation was made in Gazebo (version 10.2.0), a 3D dynamic simulator made by the Open Source Robotics Foundation, which provides a robust physics engine (to simulate gravity, inertia, contacts, etc.) in a high-quality graphics

environment. Concurrently with this work, a senior design team (which I was a part of) proposed a more refined swerve steer module design that features suspension, hub motors and more accurate encoders. The suspension is intended to help the robot overcome small obstacles and maintain reliable traction on dusty and uneven floors. The use of hub motors provides eliminates limitations in actuation accuracy associated with gear backlash in the drive gear train on the original AndyMark modules. More accurate encoders allow for accurate and consistent feedback. In addition, it must be noted that there still remains no steer axis offset on this design, allowing the control algorithm to work effectively. This new design was used in our simulation environment because it not only has newer design features that expand on current capabilities but also because more accurate physical properties could be derived easily for the associated moving parts within the robot design. A model of robot was made using the Simulation Description Format (SDF) XML format. In the SDF model file, the robot was defined as separate SDF "links" and "joints" in a manner that allowed for vertical prismatic suspension motion, as well as drive rotation and steer rotation on all four wheels. With the physical properties (mass and inertia) inputted into the SDF model, the damping and spring stiffness of the suspension prismatic joint was tuned to the desired suspension response.

The control code was parcelled into various Robot Operating System (ROS) nodes. ROS is a set of software libraries and tools that allow for facile package management and communication between separate software processes. The backbone of ROS's modularity and fault tolerance is the implementation of "nodes" and "topics." ROS nodes are separate processes that perform individual computation tasks. Nodes

subscribe and publish messages to ROS topics, which are named buses over which nodes can communicate or exchange messages. The ROS framework for the system under study in this work can be seen in the ROS computation graph shown in Figure 4.6, which is a graph showing the inter-connectivity of the developed ROS nodes (depicted as ovals) and topics (depicted as rectangles). As seen in the graph, the entire closed-loop implementation has four ROS nodes and five ROS topics. In the list below we briefly highlight the functions of each of the four ROS nodes:

- */pid_controller*: Executes two Proportional, Integral and Derivative (PID) controllers, one for the steer and another for the drive functionality of each of the four wheel modules. It subscribes to the "*/encoder_data*" topic, which provides constantly updating steer and drive motor encoder data, and publishes desired steer and drive velocity commands to the "*/pid_commands*" topic. In total, this node runs eight low-level PID controllers.

- */gazebo_client*: Subscribes to the "*/pid_commands*" topic, which provides it with the desired drive and steer commands. As a custom C++ Gazebo plugin, this node executes these desired commands on the drive and steer joints of the robot simulation. It then publishes the updated global robot position and orientation to the "*/tracking*1" topic.

- */setpt_gen*: Defines and publishes a trajectory as a function of time (t), position (x, y), orientation (θ), velocity (v_x, v_y) and angular velocity (ω), i.e. $f(t, x, y, \theta, v_x, v_y, \omega)$.

- */controller*: Implements the control algorithm outlined in previous Chapter. It subscribes the "*/tracking*1" topic, which provides it with the global robot position and orientation feedback, and the "*/setpt*" topic, which provides the desired trajectory. The outputted control commands are published to the "*/control_tower*" topic.

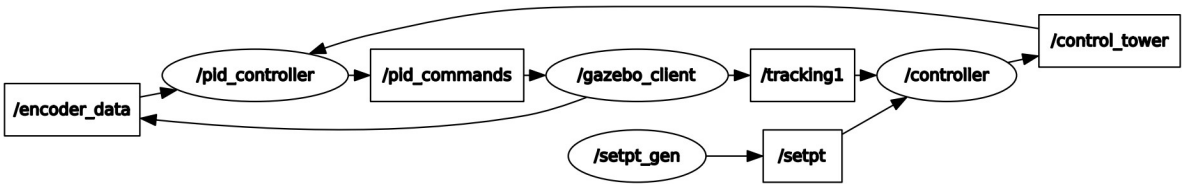


Figure 4.6: ROS Computation graph. Nodes are shown as ovals and topics are shown as rectangles.

The */setpt_gen* node generates polynomial via point trajectories defined by n via points as well as the start, T_1, and end, T_n times. As detailed by Lynch in his textbook, Modern Robotics [12], at each via point, both the desired position, $p(T_i) = p_i$ (where $i \in \{1, ..., k\}$ and p_i is the desired position at via point i), and the desired velocity, $p(\dot{T}_i) = \dot{p}_i$ (where $\dot{p}_i$ is the desired velocity at via point i) are defined. Each segment, j, (where $j \in \{1, ..., k-1\}$ in the trajectory is written as a cubic polynomial,

$$p(T_j + \delta t) = a_{j0} + a_{j1}\delta t + a_{j2}\delta t^2 + a_{j3}\delta t^3 \tag{4.8}$$

where δt is the time elapsed between via points. This polynomial is subject to the following constraints,

$$p(T_j) = p_j \qquad \dot{p}(T_j) = \dot{p}_j$$

$$p(T_j + \delta T_j) = p_{j+1} \qquad \dot{p}(T_j + \delta T_j) = \dot{p}_{j+1}$$

Solving this polynomial with the constrained outlined above, we can easily solve for $a_{j0}, ..., a_{j3}$,

$$a_{j0} = p_j \tag{4.9}$$

$$a_{j1} = \dot{p}_j \tag{4.10}$$

$$a_{j2} = \frac{3p_{j+1} - 3p_j - 2\dot{p}_j\delta T_j - \dot{p}_{j+1}\delta T_j}{\delta T_j^2} \tag{4.11}$$

$$a_{j3} = \frac{2p_j + (\dot{p}_j + \dot{p}_{j+1})\delta T_j - 2p_{j+1}}{\delta T_j^3} \tag{4.12}$$

The code for this ROS framework and the Gazebo Environment was written in both C++ and Python. As mentioned previously, the integration of Gazebo and ROS was facilitated by the development of a custom Gazebo plugin. The final simulation environment with the robot model is shown in Figure 4.7.

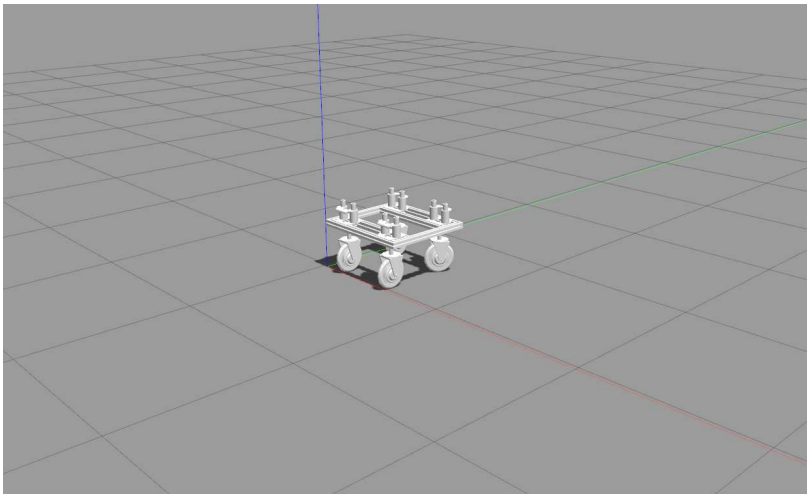


Figure 4.7: Gazebo simulation environment with model of Scrat

4.2.2 Results

The simulations presented in this section were done using a Lenovo Thinkpad P50 with a 2.80 GHz Intel Xeon(R) E3-1505M CPU, NVIDIA Quadro M2000M graphics card, 16GB DDR4 RAM, and Ubuntu Kinetic (16.04) Operating System.

Figure 4.8 shows the robot executing a simple L-shaped trajectory without changing its orientation. The trajectory tracking response is shown in Figure 4.9. It is evident that the robot can very accurately track the desired motion (shown as dotted lines) with negligible error.

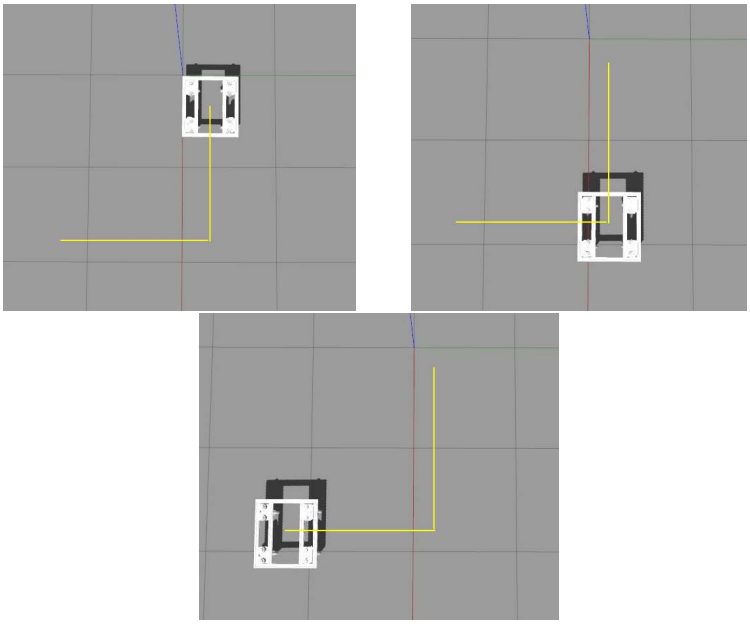


Figure 4.8: L-shape trajectory without orientation change

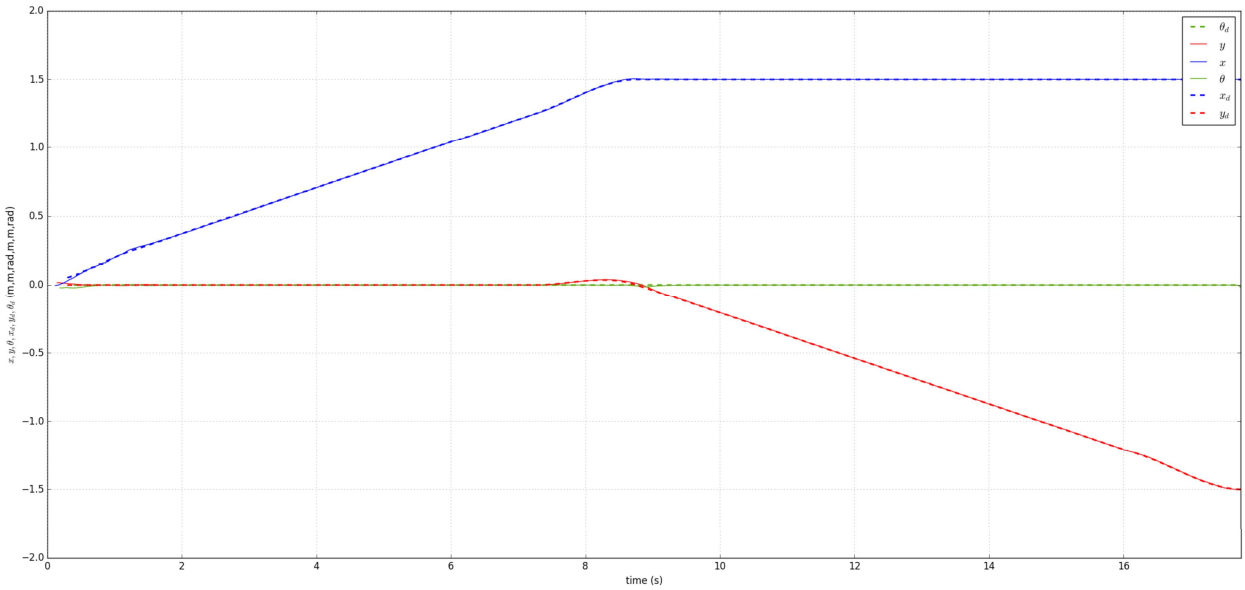


Figure 4.9: L-shape trajectory tracking response. x, y, and θ of the robot are colored blue, red and green respectively. The associated desired trajectories for x, y, and θ are in the same colors and shown as dashed lines.

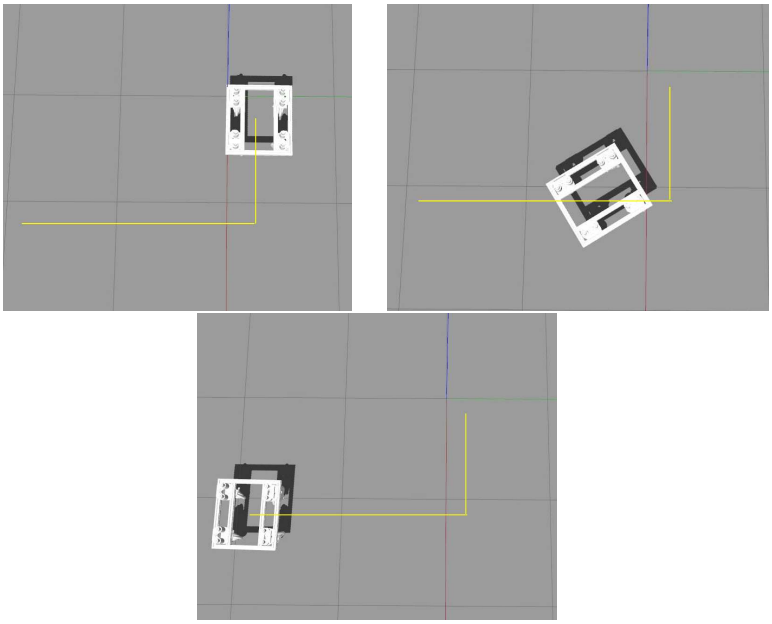


Figure 4.10: L-shape trajectory without orientation change

Figure 4.10 shows the robot executing a more complicated L-shaped trajectory that involves changing its orientation simultaneously. In this case, the orientation changes by π radians over the L-shaped path. The trajectory tracking response is shown in Figure 4.11. Once again, the robot can effectively tracks the desired motion (shown as dotted lines) with minimal error.

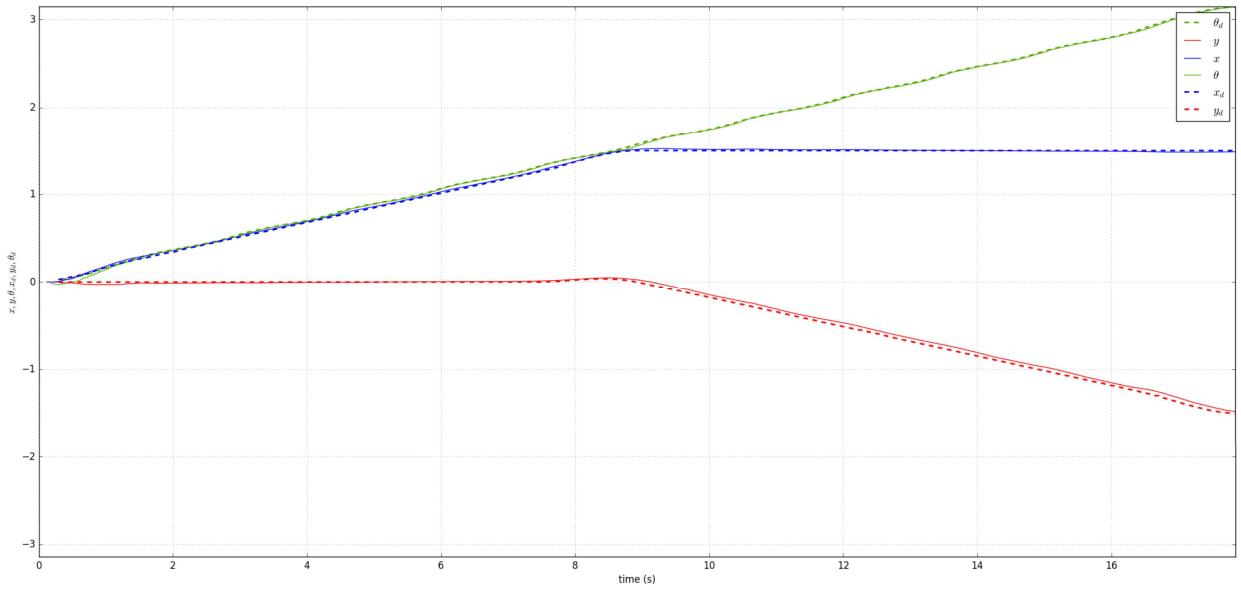


Figure 4.11: L-shape trajectory with π rad orientation change tracking response. x, y, and θ of the robot are colored blue, red and green respectively. The associated desired trajectories for x, y, and θ are in the same colors and shown as dashed lines.

Chapter 5

Conclusion

In this work, we discussed the modeling and control problem for a wheeled mobile robot (WMR) with centered orientable conventional wheels. First, we presented the derivation of an instantaneous center of rotation (ICR)-based kinematic model that defines the robot x, y velocities and angular rotation with respect to the x, y velocities at the steer axis of a chosen 'master' wheel. This model is not subject to the $\omega = 0$ or the pure translation singularity that is commonly observed with ICR-based modeling. We then rewrote this kinematic model in state-space form and determined that the system is controllable for all $\boldsymbol{x}$. Following this proof, we proposed a novel master-slave control structure that solves the nonlinear control problem for the 'master' wheel kinematic model, and coordinates the drive speeds and steer angles of the other 'slave' wheels such that their drive axes always intersect at the robot's ICR. This control structure makes the control problem for such WMRs more tractable. We then proposed feedback linearization as a nonlinear control solution for the system.

Our modeling and control technique was then simulated in MATLAB. The resulting time-response plots for the states, control input, as well as the drive speeds and steer angles verify the validity of this approach. Furthermore, despite the well-known difficulty in setting saturation limits on the feedback linearized control input,

a method to define drive speed saturation limits was developed to make this work's proposed control structure feasible for hardware.

While the control structure proposed in this work is defined specifically for a WMR with four centered orientable conventional (COC) wheels, it is easily extendable to systems with n-COC wheels. Future work will include implementing the proposed method on our robot and evaluating its performance. To cope with model uncertainty, it may be necessary to combine feedback linearization with robust or adaptive control. We also plan on generalizing the idea of ICR-based master-slave kinematic control and modeling to include WMRs with all wheel types.

Bibliography

[1] G. Campion, G. Bastin, and B. Dandrea-Novel. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1):47–62, Feb 1996.

[2] G. Campion and W. Chung. *Handbook of Robotics*, chapter Wheeled Robots. Springer, Berlin, 2008.

[3] C. P. Connette, C. Parlitz, M. Hagele, and A. Verl. Singularity avoidance for over-actuated, pseudo-omnidirectional, wheeled mobile robots. In *2009 IEEE International Conference on Robotics and Automation*, pages 4124–4130, May 2009.

[4] C. P. Connette, A. Pott, M. Hagele, and A. Verl. Control of an pseudo-omnidirectional, non-holonomic, mobile robot based on an icm representation in spherical coordinates. In *2008 47th IEEE Conference on Decision and Control*, pages 4976–4983, Dec 2008.

[5] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger. Singularity avoidance for nonholonomic, omnidirectional wheeled mobile platforms with variable footprint. In *2011 IEEE International Conference on Robotics and Automation*, pages 6136–6142, May 2011.

[6] J. K. Hedrick and A. Girard. *Control of Nonlinear Dynamic Systems: Theory and Applications*. Self-published, 2010.

[7] Viktor Kalman. *On modeling and control of omnidirectional wheels*. PhD thesis, Budapest University of Technology and Economics, Budapest, 2013.

[8] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 500–505, March 1985.

[9] A. Kulkarni. *Instant Center Based Kinematic and Dynamic Motion Synthesis for Planar Mobile Platforms*. PhD thesis, Univ. of Texas at Austin, Austin, Texas, 2009.

[10] G. Campion L. Ferriere, B. Raucent. Universite catholique de louvain. In *2015 IEEE Int. Conference on Robotics and Automation*, Minneapolis, 1996.

[11] Chuntao Leng, Qixin Cao, and Yanwen Huang. A motion planning method for omnidirectional mobile robot based on the anisotropic characteristics. *International Journal of Advanced Robotic Systems*, 5(4):45, 2008.

[12] Kevin M Lynch and Frank C Park. *Modern Robotics*. Cambridge University Press, 2017.

[13] Z. Dewey M. Pitsch, R. Anderson and M. Pryor. Self-navigating robotic assistant for long-term wide area floor contamination monitoring. In *Proceedings of the Waste Management Symposium*, Phoenix, Arizona, 2018.

[14] Lupiaez R et al. Moreno J, Clotet E. Design, implementation and validation of the three-wheel holonomic motion system of the assistant personal robot (apr). *Sensors (Basel)*, 16(10):1658, 2016.

[15] Patrick F. Muir and Charles P. Neuman. *Autonomous Robot Vehicles*. Springer New York, New York, NY, 1990.

[16] B. Thuilot, B. d'Aandrea-Novel, and A. Micaelli. Modeling and feedback control of mobile robots equipped with several steering wheels. *IEEE Transactions on Robotics and Automation*, 12(3):375–390, June 1996.